

update the VCS Canvas.

Example of Use:

```
a=vcs.init()
```

To Create a new instance of isofill use:

```
iso=a.createisofill('new','quick') # Copies content of 'quick'  
iso=a.createisofill('new')        # Copies content of 'default'
```

To Modify an existing isofill use:

```
iso=a.getisofill('AMIP_psl')  
  
iso.list()                                # Will list all the isofill  
iso.projection='linear'  
lon30={-180:'180W',-150:'150W',0:'Eq'}  
iso.xticlabels1=lon30  
iso.xticlabels2=lon30  
iso.xticlabels(lon30, lon30)              # Will set them both  
iso.xmtics1=''  
iso.xmtics2=''  
iso.xmtics(lon30, lon30)                  # Will set them both  
iso.yticlabels1=lat10  
iso.yticlabels2=lat10  
iso.yticlabels(lat10, lat10)             # Will set them both  
iso.ymtics1=''  
iso.ymtics2=''  
iso.ymtics(lat10, lat10)                  # Will set them both  
iso.datawc_y1=-90.0  
iso.datawc_y2=90.0  
iso.datawc_x1=-180.0  
iso.datawc_x2=180.0  
iso.datawc(-90, 90, -180, 180)          # Will set them all  
xaxisconvert='linear'  
yaxisconvert='linear'  
iso.xyscale('linear', 'area_wt')        # Will set them both  
missing=241                               # Color index value range 0  
  
iso.legend=None
```

There are two possibilities for setting the isofill levels:

A) Levels are all contiguous (Examples):

```
iso.levels=( [0,20,25,30,35,40], )  
iso.levels=( [0,20,25,30,35,40,45,50] )  
iso.levels=[0,20,25,30,35,40]  
iso.levels=(0.0,20.0,25.0,30.0,35.0,40.0,50.0)
```

B) Levels are not contiguous (Examples):

```
iso.levels=( [0,20], [30,40], [50,60] )  
iso.levels=( [0,20,25,30,35,40], [30,40], [50,60] )
```

There are three possibilities for setting the fillarea color ind

```
iso.fillareacolors=( [22,33,44,55,66,77] )  
iso.fillareacolors=(16,19,33,44)  
iso.fillareacolors=None
```

There are three possibilities for setting the fillarea style (Ex)

```
iso.fillareastyle = 'solid'
iso.fillareastyle = 'hatch'
iso.fillareastyle = 'pattern'
```

There are two ways to set the fillarea hatch or pattern indices

```
iso.fillareaindices=(1,3,5,6,9,20])
iso.fillareaindices=(7,1,4,9,6,15)
See using fillarea objects below!
```

Using the fillarea secondary object (Ex):

```
f=createfillarea('fill1')
To Create a new instance of fillarea use:
fill=a.createisofill('new','quick') # Copies 'quick'
fill=a.createisofill('new') # Copies 'default' to

To Modify an existing isofill use:
fill=a.getisofill('def37')

iso.fillareaindices=(7,fill,4,9,fill,15) # Set index
fill.list() # list fill
fill.style='hatch' # change style
fill.color=241 # change color
fill.index=3 # change index
```

```
ext_1='n'
ext_2='y'
iso.exts('n', 'y' ) # Will set them both
```

Methods defined here:

```
__init__(self, parent, Gfi_name=None, Gfi_name_src='default', createGfi=0)
```

```
datawc(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)
```

```
exts(self, ext1='n', ext2='y')
```

```
list(self)
```

```
rename = renameGfi(self, old_name, new_name)
```

```
#####
#
# Function:      renameGfi
#
# Description of Function:
#     Private function that renames the name of an existing
#     graphics method.
#
#
# Example of Use:
```

```

#         renameGfi(old_name, new_name)
#             where: old_name is the current name of isofill
#                   new_name is the new name for the isofill
#
#####

```

script(self, script_filename=None, mode=None)

Function: script # Calls _vcs.s

Description of Function:

Saves out a isofill graphics method in Python or VCS script to a designated file.

Example of Use:

```
script(scriptfile_name, mode)
```

where: scriptfile_name is the output name of the file
mode is either "w" for replace or "a" for append

Note: If the the filename has a ".py" at the end it will produce a Python script. If the filename has a ".scr" it will produce a VCS script. If neither extension is present, default a Python script will be produced.

```

a=vcs.init()
iso=a.createisofill('temp')
iso.script('filename.py') # Append to a Python file
iso.script('filename.scr') # Append to a VCS file
iso.script('filename','w')

```

xmtics(self, xmt1=",", xmt2="")

xticlabels(self, xtl1=",", xtl2="")

xyyscale(self, xat=",", yat="")

ymtics(self, ymt1=",", ymt2="")

yticlabels(self, ytl1=",", ytl2="")

Properties defined here:

datawc_calendar

```

get">get = _getcalendar(self)
set">set = _setcalendar(self, value)

```

datawc_timeunits

```

get">get = _gettimeunits(self)
set">set = _settimeunits(self, value)

```

datawc_x1

```

get">get = _getdatawc_x1(self)
set">set = _setdatawc_x1(self, value)

```

datawc_x2

```
get">get = _getdatawc_x2(self)
set">set = _setdatawc_x2(self, value)
```

datawc_y1

```
get">get = _getdatawc_y1(self)
set">set = _setdatawc_y1(self, value)
```

datawc_y2

```
get">get = _getdatawc_y2(self)
set">set = _setdatawc_y2(self, value)
```

ext_1

```
get">get = _gettext_1(self)
set">set = _settext_1(self, value)
```

ext_2

```
get">get = _gettext_2(self)
set">set = _settext_2(self, value)
```

fillareacolors

```
get">get = _getfillareacolors(self)
set">set = _setfillareacolors(self, value)
```

fillareaindices

```
get">get = _getfillareaindices(self)
set">set = _setfillareaindices(self, value)
```

fillareastyle

```
get">get = _getfillareastyle(self)
set">set = _setfillareastyle(self, value)
```

legend

```
get">get = _getlegend(self)
set">set = _setlegend(self, value)
```

levels

```
get">get = _getlevels(self)
set">set = _setlevels(self, value)
```

missing

```
get">get = _getmissing(self)
set">set = _setmissing(self, value)
```

name

```
get">get = _getname(self)
set">set = _setname(self, value)
```

projection

```
get">get = _getprojection(self)
set">set = _setprojection(self, value)
```

xaxisconvert

```
get">get = _getxaxisconvert(self)  
set">set = _setxaxisconvert(self, value)
```

xmtics1

```
get">get = _getxmtics1(self)  
set">set = _setxmtics1(self, value)
```

xmtics2

```
get">get = _getxmtics2(self)  
set">set = _setxmtics2(self, value)
```

xticlabels1

```
get">get = _getxticlabels1(self)  
set">set = _setxticlabels1(self, value)
```

xticlabels2

```
get">get = _getxticlabels2(self)  
set">set = _setxticlabels2(self, value)
```

yaxisconvert

```
get">get = _getyaxisconvert(self)  
set">set = _setyaxisconvert(self, value)
```

ymtics1

```
get">get = _getymtics1(self)  
set">set = _setymtics1(self, value)
```

ymtics2

```
get">get = _getymtics2(self)  
set">set = _setymtics2(self, value)
```

yticlabels1

```
get">get = _getyticlabels1(self)  
set">set = _setyticlabels1(self, value)
```

yticlabels2

```
get">get = _getyticlabels2(self)  
set">set = _setyticlabels2(self, value)
```

Data and other attributes defined here:

```
__slots__ = ['setmember', 'parent', 'name', 'g_name', 'xaxisconvert', 'yaxisconvert', 'levels', 'fillareacolor',  
'fillareaindices', 'ext_1', 'ext_2', 'missing', 'projection', 'xticlabels1', 'xticlabels2', 'yticlabels1', 'yticlabels2',  
'xmtics2', ...]
```

```
g_name = <member 'g_name' of 'Gfi' objects>
```

```
parent = <member 'parent' of 'Gfi' objects>
```

```
setmember = <member 'setmember' of 'Gfi' objects>
```

Functions

```
add_level_ext_1(self, ext_value)
#####
#
# Function:      add_level_ext_1
#
# Description of Function:
#     Private function that adds the extension triangle to the
#     legend on the plot
#
#
# Example of Use:
#     add_level_ext_1(self, ext_value)
#         where: self is the class (e.g., Gfi)
#         ext_value is either 'n' to remove the triangle
#         legend or 'y' to show the triangle on the
#
#####
```

```
add_level_ext_2(self, ext_value)
#####
#
# Function:      add_level_ext_2
#
# Description of Function:
#     Private function that adds the extension triangle to the
#     legend on the plot
#
#
# Example of Use:
#     add_level_ext_2(self, ext_value)
#         where: self is the class (e.g., Gfi)
#         ext_value is either 'n' to remove the triangle
#         legend or 'y' to show the triangle on the
#
#####
```

```
getGfimember(self, member)
#####
#
# Function:      getGfimember
#
# Description of Function:
#     Private function that retrieves the isofill members from the
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
```

```

#         getGfimember(self,name)
#             where: self is the class (e.g., Gfi)
#                   name is the name of the member that is being
#
#####

```

getmember = getGfimember(self, member)

```

#####
#
# Function:         getGfimember
#
# Description of Function:
#     Private function that retrieves the isofill members from t
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#         getGfimember(self,name)
#             where: self is the class (e.g., Gfi)
#                   name is the name of the member that is being
#
#####

```

renameGfi(self, old_name, new_name)

```

#####
#
# Function:         renameGfi
#
# Description of Function:
#     Private function that renames the name of an existing isof
#     graphics method.
#
#
# Example of Use:
#         renameGfi(old_name, new_name)
#             where: old_name is the current name of isofill grap
#                   new_name is the new name for the isofill gra
#
#####

```

setGfimember(self, member, value)

```

#####
#
# Function:         setGfimember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:

```

```

#      setGfimember(self,name,value)
#              where: self is the class (e.g., Gfi)
#              name is the name of the member that is being
#              value is the new value of the member (or att
#
#####

setmember = setGfimember(self, member, value)
#####
#
# Function:      setGfimember
#
# Description of Function:
#      Private function to update the VCS canvas plot. If the can
#      set to 0, then this function does nothing.
#
#
# Example of Use:
#      setGfimember(self,name,value)
#              where: self is the class (e.g., Gfi)
#              name is the name of the member that is being
#              value is the new value of the member (or att
#
#####

```

Data

StringTypes = (<type 'str'>, <type 'unicode'>)